**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

**КУРСОВОЙ ПРОЕКТ**

По дисциплине МДК.07.01 Управление и автоматизация баз данных

**Разработка базы данных**

**«Автосалон»**

Выполнил студент гр. ИСП-20 /\_\_\_\_\_\_\_\_\_\_\_\_/ Жуков Алексей Николаевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ**

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_/ Егоров Данил Павлович/

(подпись) (Ф.И.О.)

Воркута

2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc153011997)

[1.1 СУБД SQL Server 4](#_Toc153011998)

[1.2 Microsoft SQL Server Management Studio 5](#_Toc153011999)

[1.3 Entity Framework 6](#_Toc153012000)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 7](#_Toc153012001)

[2.1 Разработка диаграммы ERD 7](#_Toc153012002)

[2.2 Разработка базы данных 9](#_Toc153012003)

[2.3 Словарь данных 10](#_Toc153012004)

[ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ 13](#_Toc153012005)

[3.1 Введение в безопасность SQL Server 13](#_Toc153012006)

[3.2 Управление безопасностью уровня сервера 14](#_Toc153012007)

[3.3 Управление участниками уровня базы данных 15](#_Toc153012008)

[3.4 Управление разрешениями уровня базы данных 24](#_Toc153012009)

[3.5 Шифрование данных баз данных 26](#_Toc153012010)

[ЗАКЛЮЧЕНИЕ 28](#_Toc153012011)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 29](#_Toc153012012)

# ВВЕДЕНИЕ

В современном мире автомобильные технологии играют значительную роль в повседневной жизни людей. В связи с этим, автосалоны, предоставляющие услуги по продаже и обслуживанию автомобилей, становятся все более востребованными. Для эффективной работы таких предприятий необходимо использование современных информационных систем, которые обеспечат удобный контроль над каталогом и учет продаж. Следовательно, создание распределенной информационной системы. Развитие автобизнеса во всем мире на современном этапе зачастую представляется практически невозможным без внедрения систем автоматизации.

**Объект** – база данных система «Автосалон»

**Предмет** – автоматизация бизнес-процесса учёта автомобилей системы «Автосалон»

**Цель** – создание базы данных для информационной системы «Автосалон»

**Задачи:**

* выбрать инструментарий к информационной системе «Автосалон»;
* спроектировать базу данных к информационной системе «Автосалон»;
* сделать разграничение ролей для базы данных;
* сделать шифрование данных для базы данных;
* разработать информационную систему «Автосалон».

**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

## 1.1 СУБД SQL Server

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language). Клиент (например, внешняя программа) отправляет запрос на языке SQL посредством специального API. СУБД должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

В зависимости от задачи, которую выполняет команда T-SQL, он может принадлежать к одному из следующих типов:

DDL (Data Definition Language / Язык определения данных). К этому типу относятся различные команды, которые создают базу данных, таблицы, индексы, хранимые процедуры и т.д. В общем определяют данные.

В частности, к этому типу мы можем отнести следующие команды:

* CREATE: создает объекты базы данных (саму базу данных, таблицы, индексы и т.д.)
* ALTER: изменяет объекты базы данных
* DROP: удаляет объекты базы данных
* TRUNCATE: удаляет все данные из таблиц

DCL (Data Control Language / Язык управления доступа к данным). К этому типу относят команды, которые управляют правами по доступу к данным. В частности, это следующие команды:

* GRANT: предоставляет права для доступа к данным
* REVOKE: отзывает права на доступ к данным

## 1.2 Microsoft SQL Server Management Studio

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server, Базы данных SQL Azure, Управляемого экземпляра SQL Azure, SQL Server на виртуальной машине Azure и Azure Synapse Analytics. Среда SSMS предоставляет единую комплексную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом многофункциональных редакторов скриптов для доступа к SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

Среда SQL Server Management Studio предназначена для доступа к службам Службы Analysis Services, Integration Services и Reporting Services, а также для их настройки, администрирования и управления ими. Хотя все три технологии бизнес-аналитики полагаются на среду SQL Server Management Studio, административные задачи, связанные с каждой из этих технологий, несколько отличаются.

Среда SQL Server Management Studio позволяет управлять объектами Службы Analysis Services, например выполнять их резервное копирование и обработку.

Среда Management Studio позволяет создавать проекты скриптов служб Службы Analysis Services, в которых выполняются разработка и сохранение скриптов с использованием многомерных выражений (MDX), расширений интеллектуального анализа данных (DMX) и XML для аналитики (XMLA). Проекты скриптов служб Службы Analysis Services используются для выполнения задач управления или повторного создания баз данных, кубов и других объектов в экземплярах служб Службы Analysis Services. Например, можно разработать скрипт XMLA в проекте скрипта служб Службы Analysis Services, который создает объекты непосредственно в существующем экземпляре служб Службы Analysis Services. Проекты скриптов служб Службы Analysis Services могут быть сохранены в составе решения и интегрироваться с контролем исходного кода.

## 1.3 Entity Framework

Entity Framework — это решение для работы с базами данных, которое используется в программировании на языках семейства .NET. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), а не таблиц. Также код с использованием EF пишется гораздо быстрее.

Например, работая с базами данных напрямую, разработчик должен беспокоиться о подключении, подготовке SQL и параметров, отправке запросов и транзакций. На Entity Framework всё это делается автоматически — программист же работает непосредственно с сущностями и только говорит EF, что нужно сохранить изменения.

В этой статье будут разобраны основы применения Entity Framework, для понимания которых нужно владеть ADO.NET — пользоваться базами данных, писать SQL-запросы и работать с подключениями.

Entity Framework позволяет значительно сократить код работы с базами данных. При этом он предоставляет большие возможности. Например, можно использовать:

* foreign keys;
* связи one-to-one, one-to-many и many-to-many;
* параметризацию запросов;
* хранимые процедуры.

Однако стоит учитывать, что EF выступает прослойкой между приложением и базой данных, поэтому может ухудшаться производительность. Для небольших проектов это допустимо, но если программа должна работать под большой нагрузкой, то лучше использовать чистый ADO.NET.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

## 2.1 Разработка диаграммы ERD

Данная диаграмма — (ER-модель данных) обеспечивает стандартный способ определения данных и отношений между ними. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Диаграммы «сущность— связь» в отличие от функциональных диаграмм определяют спецификации структур данных программного обеспечения.

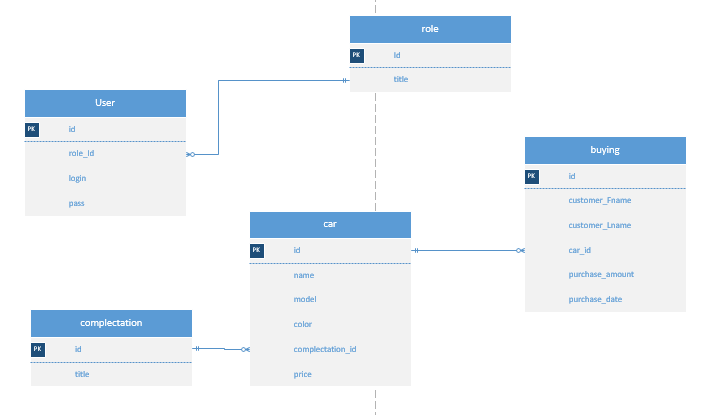
Базовыми понятиями ER-модели данных (ER — Entity— Relationship) являются сущность, атрибут и связь. Сущность — это класс однотипных реальных или абстрактных объектов (людей, событий, состояний, предметов и т.п.), информация о которых имеет существенное значение для рассматриваемой предметной области.

Атрибут — любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут, таким образом, представляет собой некоторый тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов.

Атрибуты делятся на ключевые, т. е. входящие в состав уникального идентификатора ключа, и описательные — прочие.

Первичный ключ — это атрибут или совокупность атрибутов и связей, предназначенная для уникальной идентификации каждого экземпляра сущности.

Описательные атрибуты могут быть обязательными или необязательными. Обязательные атрибуты для каждой сущности всегда имеют конкретное значение, необязательные могут быть не определены.

Связь — это отношение одной сущности к другой или к самой себе. Если любой экземпляр одной сущности связан хотя бы с одним экземпляром другой сущности, то связь является обязательной. Необязательная связь представляет собой условное отношение между сущностями. Каждая сущность может обладать любым количеством связей с другими сущностями модели. Различают три типа отношений «один-к-одному»; «один-ко-многим»; «многие-ко-многим».

*Рисунок 2.1 ER Диаграмма*

Основными сущностями для решения указанной задачи являются: user, buying. Сразу возникает очевидная связь между сущностями user и role для авторизации.

Проанализируем атрибуты сущностей. buiing может иметь имя и фамилию покупателя, информация о машине, сумма покупки и дата покупки, у каждой сущности есть ID. Каждый новый отчёт имеет номер.

## 2.2 Разработка базы данных

Создание базы данных.

В обозревателе объектов подключитесь к экземпляру компонента Компонент SQL Server Database Engine и разверните его.

Щелкните правой кнопкой мыши узел Базы данных и выберите команду Создать базу данных.

В поле Новая база данных введите имя базы данных.

Чтобы создать базу данных, приняв все значения по умолчанию, нажмите кнопку ОК; в противном случае продолжайте выполнять указанные ниже дополнительные действия.

Чтобы изменить имя владельца, нажмите ( … ) и выберите другого владельца.

Чтобы изменить значения первичных данных по умолчанию и файлы журнала транзакций, выберите соответствующую ячейку в сетке Файлы базы данных и введите новое значение. Дополнительные сведения см. в статье Добавление файлов данных или журналов в базу данных.

Чтобы изменить параметры сортировки базы данных, выберите страницу Параметры и выберите из списка желаемые параметры сортировки.

Чтобы изменить модель восстановления, выберите страницу Параметры и модель восстановления из списка.

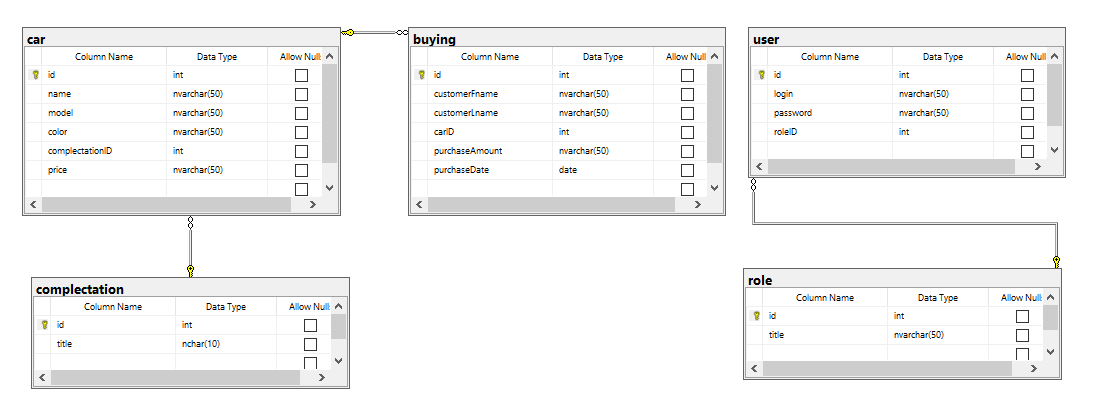
Чтобы изменить параметры базы данных, выберите страницу Параметры и измените параметры базы данных. Описание каждого параметра см. в разделе Параметры ALTER DATABASE SET (Transact-SQL).

Чтобы добавить новую файловую группу, перейдите на страницу Файловые группы. Нажмите кнопку Добавить и введите значения для файловой группы.

Чтобы добавить расширенное свойство в базу данных, выберите страницу Расширенные свойства.

В столбце Имя введите имя расширенного свойства.

В столбце Значение введите текст расширенного свойства. Например, введите одно или несколько предложений, которые описывают базу данных.

Чтобы создать базу данных, нажмите кнопку ОК.

*Рисунок 2.2 База данных приложения*

Здесь 5 таблиц, есть user и role – они отвечают за авторизацию в приложении, car – это информация о машинах, complectation – вспомогательный для информации о комплектации автомобиля, buying – информация о покупках.

## 2.3 Словарь данных

Словарь данных - это структурированное описание данных, которое содержит информацию о структуре, организации и свойствах данных в базе данных. Он обычно используется для документирования метаданных, то есть информации о данных в базе данных. В словаре данных обычно содержатся следующие элементы:

1. Описание таблиц и полей: словарь данных содержит информацию о различных таблицах в базе данных, их названия, структуру, связи с другими таблицами, и описание каждого поля в таблице.

2. Описание ограничений и связей: словарь данных может также содержать информацию о различных ограничениях целостности данных, связях между таблицами и других ограничениях, которые определяют правила и условия для хранения данных.

3. Описания процедур и функций: метаданные о реализованных программных объектах, таких как хранимые процедуры, функции, триггеры, представления и т.д., могут быть также включены в словарь данных.

Словарь данных является важным инструментом для анализа и понимания структуры базы данных. Он помогает администраторам и разработчикам понять, как организованы данные, и обеспечить их правильное использование и обслуживание.

Таблица 1 Роль

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **role** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| P/KEY | **id** | int | Y | Уникальный идентификатор |
|  | **title** | nvarchar(50) | Y | Название роли |

Таблица 2 Пользователь

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **userlog** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| P/KEY | **id** | int | Y | Уникальный идентификатор |
|  | **login** | nvarchar(50) | Y | Название роли |
|  | **password** | nvarchar(50) | Y | Пароль |
| F/KEY | **roleID** | int | Y | Уникальный идентификатор  роли |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **car** | | | | |
| **KEY** | **FIELD**  **NAME** | **DATA TYPE /FIELD SIZE** | **REQUIRED?** | **NOTES** |
| P/KEY | **id** | int | Y | Уникальный идентификатор |
|  | **name** | nvarchar(50) | Y | Название марки |
|  | **model** | nvarchar(50) | Y | Модель машины |
|  | **color** | nvarchar(50) | Y | Цвет машины |
| F/KEY | **competitionID** | int | Y | Уникальный идентификатор комплектации |
|  | **price** | Nvarchar(50) | Y | Цена машины |

Таблица 3 Машина

Таблица 4 Комплектация

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **complectation** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| P/KEY | **id** | int | Y | Уникальный идентификатор |
|  | **title** | nvarchar(10) | Y | Название комплектации |

Таблица 5 Покупки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **buying** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| P/KEY | **id** | int | Y | Уникальный идентификатор |
|  | **customerFname** | nvarchar(50) | Y | Имя покупателя |
|  | **customerLname** | nvarchar(50) | Y | Фамилия покупателя |
| F/KEY | **carID** | int | Y | Уникальный идентификатор машины |
|  | **purchaseAmount** | nvarchar(50) | Y | Стоимость покупки |
|  | **purchaseDate** | Date | Y | Дата покупки |

# ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ

## 3.1 Введение в безопасность SQL Server

Безопасность данных является одним из самых важных аспектов любой базы данных, включая SQL Server. Поддержание безопасности в SQL Server включает в себя ряд мероприятий, предназначенных для защиты данных от несанкционированного доступа, вмешательства и утечки информации.

При правильном создании резервных копий баз данных можно будет восстановить данные после многих видов сбоев, включая следующие:

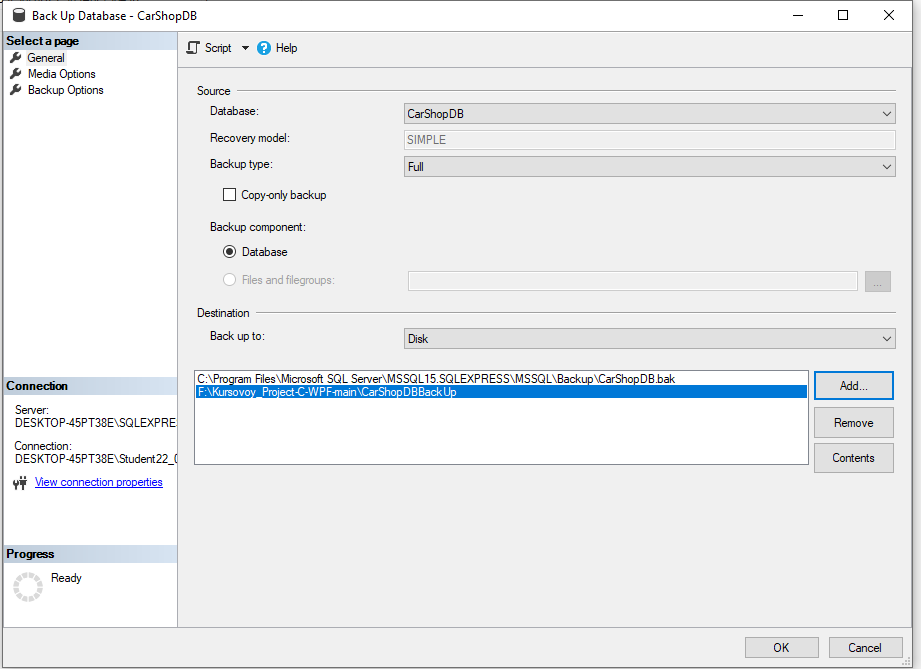
• Сбой носителя.

• ошибки пользователей (например, удаление таблицы по ошибке);

• сбои оборудования (например, поврежденный дисковый накопитель или безвозвратная потеря данных на сервере);

Для создания резервной копии базы данных с помощью SQL Server Management Studio (SSMS) следуйте этим шагам:

1. Нажмите правой кнопкой мыши на базе данных, выберите "Tasks" (Задания) и затем "Backup" (Резервное копирование).
2. В появившемся окне выберите тип резервного копирования (полное, дифференциальное, журнал транзакций).

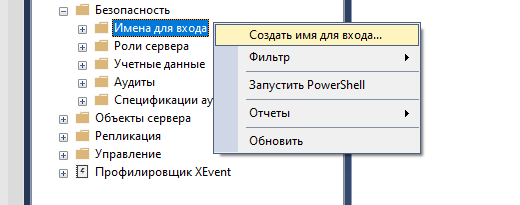


*Рисунок 3.1 Создание резервной копии*

3. Укажите путь для сохранения резервной копии базы данных.

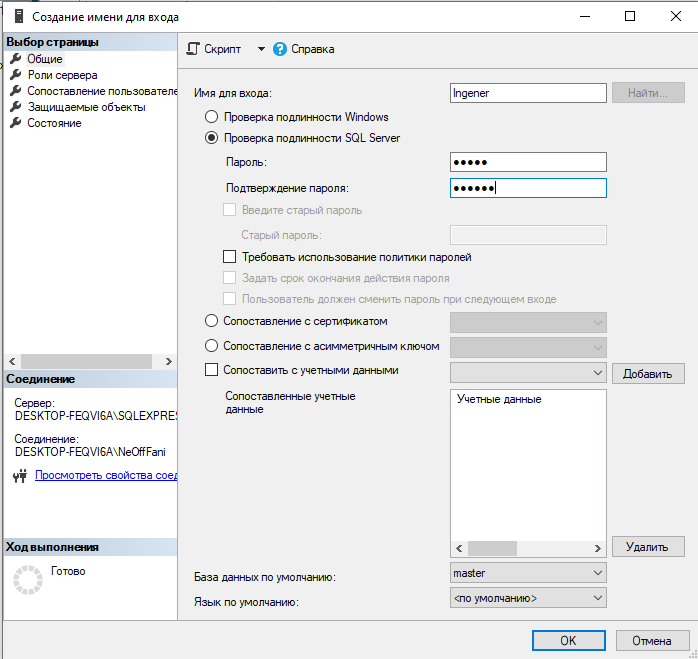
## 3.2 Управление безопасностью уровня сервера

Существует два уровня доступа к экземпляру SQL Server: учетные записи пользователя сервера (или экземпляра) и пользователи базы данных. С помощью учетных записей серверы позволяют внешнему пользователю (далее термин «пользователь» применяется для любого приложения, службы, API и т. д., пытающихся подключиться к SQL Server) выполнить начальное соединение с экземпляром SQL Server. В случае безопасности на основе SQL для этого требуются имя пользователя и пароль. В случае доверенной проверки подлинности это учетная запись домена.

Способа создать эти учетные записи пользователя через графический интерфейс. Чтобы запустить диалоговое окно для создания учетных записей пользователей, подключитесь к экземпляру SQL Server в среде SQL Server Management Studio (SSMS) в обозревателе объектов, а затем разверните узел Security\Logins («Безопасность\Имена пользователя»). Щелкните правой кнопкой мыши на пункте Logins и выберите в контекстном меню пункт New Login («Создать учетную запись»).

***Рисунок 3.2. Создание учетной записи пользователя SQL Server***

Вы увидите диалоговое окно для настройки параметров учетной записи, показанное на рисунке 3.2. Изменить имя пользователя можно в том же окне.

***Рисунок 3.3 Настройка учетной записи пользователя SQL Server***

Это вкладка General («Общие») для создания (и изменения) параметров учетной записи. На вкладке General можно задать:

**Login name («Имя пользователя»).** Используется при проверке подлинности. В случае Windows, или доверенной проверки подлинности, необходимо задать имя в формате DOMAIN\LOGIN, где LOGIN — имя пользователя внутри домена, из которого пользователь выполняет проверку подлинности. Если экземпляр SQL Server расположен в другом домене, то необходимы отношения доверия между этим доменом и доменом SQL Server.

Password («Пароль»). При проверке подлинности SQL Server текстовое поле пароля включено, и вы вводите как имя пользователя, так и связанный с ним пароль.

Password Policy («Настройки политики паролей») и Expiration («Срок действия»)***.*** Флажки для политики пароля и срока действия также установлены в режиме проверки подлинности SQL Server, и применяются те политики, которые действуют в Active Directory в домене, где размещается SQL Server. Назначая имя пользователя SQL Server, вы можете разрешить пользователям менять свои пароли после регистрации. В результате администратор базы данных лишается доступа к имени учетной записи конечного пользователя.

Перейдем к следующей вкладке Server Roles («Роли сервера»), показанной на рисунке 3.3. На этой странице можно выбрать любые роли на уровне SQL Server (экземпляра) для нового пользователя. Роли сервера представляют собой коллекции прав, также известные как защищаемые объекты, которые упаковываются в коллекцию, чтобы вам не приходилось назначать права каждому защищаемому объекту отдельно. По умолчанию каждая учетная запись является членом общедоступной роли, что позволяет установить основное подключение к экземпляру SQL Server.

|  |
| --- |
|  |
| ***Рисунок 3.4. Вкладка Server Roles*** |

Следующая страница диалогового окна Create Login в среде SQL Server Management Studio предназначена для сопоставления учетных записей пользователей. Каждая учетная запись может иметь пользователя в одной или нескольких базах данных. На этой странице можно создать пользователей базы данных, связанных с новой учетной записью. Для этого нужно предоставить следующую информацию.

Database («База данных»).Установите флажок рядом с базой данных, в которой нужно создать связанного пользователя для учетной записи.

User Name («Имя пользователя»)***.*** Имя объекта пользователя не обязательно соответствует имени учетной записи, и далее будет показано, как это можно изменить.

Database Role Membership («Членство в роли базы данных»).Точно так же, как на уровне экземпляра или сервера, каждая база данных располагает заранее определенной коллекцией прав, упакованных в ролях. Чуть позже мы рассмотрим роли базы данных, поставляемые с Microsoft SQL Server.

|  |
| --- |
|  |
| ***Рисунок 3.5. Пример настроек учетной записи пользователя*** |

На следующей странице, Securables, представлены защищаемые объекты на уровне сервера или экземпляра. Как отмечалось выше, защищаемые объекты — это разрешения, предоставленные объектам. Защищаемые объекты обычно предоставляются в следующих случаях:

предопределенная роль слишком широка (много других прав для учетной записи);

назначенная роль или набор ролей не охватывает полностью все права, необходимые для учетной записи.

|  |
| --- |
|  |
| ***Рисунок 3.6. Назначение дополнительных прав*** |

Наконец, переходим к странице Status («Состояние»). На этой странице можно разрешить или отменить доступ для пользователя (по умолчанию выбирается Grant — разрешить). Поэтому можно создать учетную запись, предоставить права, создать связанных пользователей, а затем отменить доступ. Вы можете вернуться в это окно для существующего пользователя и отменить доступ к экземпляру SQL Server. Аналогично происходит включение и отключение учетной записи (см. рисунок 3.6). Наконец, мы можем просмотреть состояние учетной записи пользователя и узнать, была ли учетная запись заблокирована из-за слишком большого числа неудачных попыток регистрации с неверным паролем.

|  |
| --- |
|  |
| ***Рисунок 3.7 Вкладка Status*** |

На данном этапе важно отметить, как организована связь пользователей базы данных с учетной записью пользователя сервера. Как я уже указывал, соответствие имен между двумя объектами необязательно. Это объясняется тем, что объекты объединены в системных таблицах не по имени, а по идентификатору, именуемому sid (идентификатор безопасности). Это позволяет избавиться от привязки к учетной записи, соответствующей имени пользователя, или избежать возникновения ситуации, в которой вы восстанавливаете базу данных с именем пользователя, например trevor на экземпляре SQL Server, где уже имеется учетная запись trevor, но это совершенно другое лицо, которое не должно иметь прав в вашей базе данных. Благодаря sid такая опасность исключается.

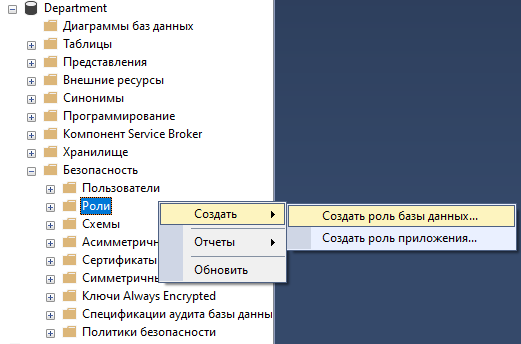
**3.3 Управление участниками уровня базы данных**

Чтобы легко управлять разрешениями в базах данных, SQL Server предоставляет несколько ролей, которые являются субъектами безопасности, которые группируют другие субъекты. Они похожи на группы в операционной системе Microsoft Windows. Разрешения ролей уровня базы данных распространяются на всю базу данных.

Существует два типа ролей уровня базы данных: предопределенные роли базы данных, являющиеся стандартными для базы данных, и пользовательские роли базы данных, которые можно создавать.

Предопределенные роли базы данных задаются на уровне базы данных и предусмотрены в каждой базе данных. Члены ролей базы данных db\_owner могут управлять членством в предопределенных ролях базы данных. Кроме того, в базе данных msdb имеются специальные роли базы данных.

Создание новой роли

В SQL Server Management Studio разверните базу данных табличной модели, для которой требуется создать новую роль, щелкните правой кнопкой мыши роли и выберите команду Создать роль.

*Рисунок 3.8. Настройка роли.*

В окне общих параметров введите имя роли в поле Имя.

Используйте имена, которые четко идентифицируют тип элемента, например, финансовые менеджеры или специалисты по персоналу, и убедитесь, что имя не содержит запятую.

В поле Разрешения базы данных для этой роли выберите один из следующих параметров разрешений.

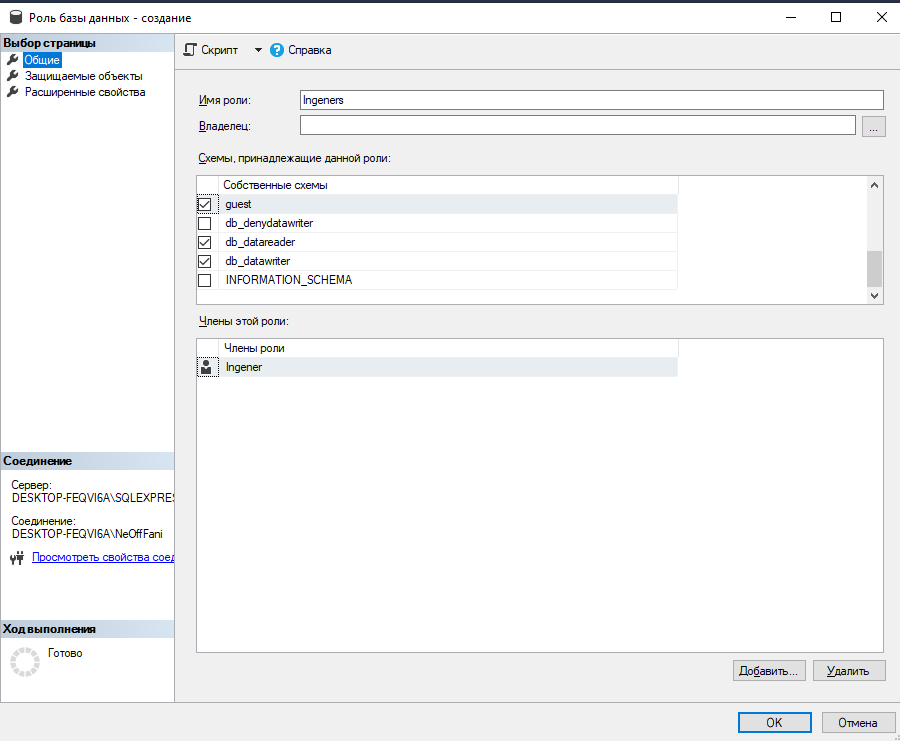
Полный доступ (администратор). Члены могут вносить изменения в схему модели, а также просматривать все данные.

Process Database. Члены могут выполнять операции «Обработать» и «Обработать все». Не могут изменять схему модели и просматривать данные.

Чтение. Члены могут просматривать данные (с учетом фильтров строк), но не могут вносить изменения в схему модели.

В диалоговом окне Создание роли в окне «Выбор страницы» щелкните Членство.

*Рисунок 3.9. Создание новой роли*

 В окне параметров членства нажмите кнопку добавить, а затем в диалоговом окне выбор пользователей или групп добавьте пользователей или группы, которые вы хотите добавить в качестве участников.

## 3.4 Управление разрешениями уровня базы данных

Соглашения об именовании разрешений

Ниже описаны общие соглашения, которые соблюдаются при задании имен разрешениям.

Производительности

Предоставляет возможности, схожие с владением. Имеющий это разрешение получает все установленные разрешения на защищаемую сущность. Участник, получивший разрешение CONTROL, может также предоставлять разрешения на защищаемую сущность другим участникам. Так как модель безопасности SQL Server является иерархической, CONTROL на определенном область неявно включает CONTROL для всех защищаемых объектов в этом область.

ИЗМЕНИТЬ

Предоставляет возможность изменения свойств определенной защищаемой сущности, кроме ее владельца. При предоставлении разрешения ALTER на ту или иную область также предоставляется возможность изменения, создания или удаления любой защищаемой сущности, содержащейся в пределах данной области.

ALTER ANY <Server Securable, где защищаемый> сервер может быть любым защищаемым сервером.

Предоставляет возможность создавать, изменять и удалять отдельные экземпляры Защищаемой сущности сервера.

ALTER ANY <Database Securable, где защищаемая> база данных может быть любой защищаемой на уровне базы данных.

Предоставляет возможность СОЗДАВАТЬ, ИЗМЕНЯТЬ и УДАЛЯТЬ отдельные экземпляры Защищаемой сущности базы данных.

TAKE OWNERSHIP

Позволяет получать во владение защищаемую сущность, на которую предоставлено разрешение.

IMPERSONATE <Имя\_для\_входа>

Позволяет олицетворять имя входа.

IMPERSONATE <Пользователь>

Позволяет олицетворять пользователя.

CREATE <Защищаемый объект сервера>

Предоставляет возможность создавать Защищаемую сущность сервера.

CREATE <Защищаемый объект базы данных>

Предоставляет возможность создавать Защищаемую сущность базы данных.

CREATE <Защищаемый объект, содержащийся в схеме>

Предоставляет возможность создавать защищаемую сущность, содержащуюся в схеме. Однако для создания защищаемой сущности в той или иной схеме на эту схему требуется разрешение ALTER.

VIEW DEFINITION

Разрешает доступ к метаданным.

ССЫЛКИ

Разрешение REFERENCES для таблицы необходимо для создания ограничения FOREIGN KEY, которое ссылается на эту таблицу.

Разрешение REFERENCES для объекта необходимо для создания FUNCTION или VIEW с предложением 2 WITH SCHEMABINDING , которое ссылается на этот объект.

## 3.5 Шифрование данных баз данных

Шифрование данных в базах данных SQL Server - это важный аспект безопасности, который позволяет защитить конфиденциальную информацию от несанкционированного доступа. В SQL Server Management Studio (SSMS) можно использовать различные методы шифрования для защиты данных.

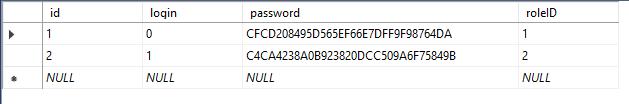
В нашей базе данных используется тип шифрования MD5. MD5 - это алгоритм хэширования, который используется для создания уникального хэш-значения из входных данных фиксированной длины. В SQL Server MD5 обычно используется как метод для создания хэш-значений, которые могут быть использованы для проверки целостности данных, а также в качестве метода необратимого шифрования паролей и других конфиденциальных данных.

Что бы зашифровать данные в SSMS понадобится небольшой запрос

UPDATE userlog

SET password = CONVERT(varchar(64),

HASHBYTES('MD5',password),2)



*Рисунок 3.10. Зашифрованные данные*

Благодаря этому запросу мы зашифровали строку с паролями пользователей

Но что бы в программе не надо было вводить зашифрованный пароль нужно реализовать расшифровку данных из базы данных так что бы вводился пароль установленный изначально.

Для этого нужно создать класс в котором будет данный код:

internal class Class1

{

public static string hashPassword(string password)

{

MD5 md5 = MD5.Create();

byte[] b = Encoding.UTF8.GetBytes(password);

byte[] hash = md5.ComputeHash(b);

StringBuilder sb = new StringBuilder();

foreach (var a in hash)

sb.Append(a.ToString("X2"));

return Convert.ToString(sb);

}

}

И теперь при входе в приложение нужно будет ввести обычный пароль в то время как в базе данных он будет зашифрован.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта по разработке базы данных для автосалона были достигнуты следующие цели: разработана структура базы данных, реализовано разграничение ролей сотрудников, обеспечена шифровка конфиденциальных данных, а также настроено резервное копирование для защиты от потери информации. Эти меры обеспечивают эффективное управление данными, обеспечивают безопасность и сохранность информации, что является важным компонентом успешной работы автосалона. В результате, можно сделать вывод, что разработанная база данных отвечает всем требованиям и готова к использованию в работе.

Таким образом, разработка базы данных для автосалона является важным шагом в обеспечении эффективного управления клиентскими данными, продажами и складским учетом. Развертывание ролевых доступов, шифрования данных и регулярного резервного копирования обеспечивает необходимую защиту и безопасность информации, что способствует повышению качества обслуживания клиентов и оптимизации бизнес-процессов в автосалоне. Полученные результаты позволяют сделать вывод о полной готовности разработанной базы данных к успешному использованию в реальных условиях работы автосалона.

Ссылка на репозиторий GitHub: <https://github.com/fankfankov/Kursovoy_Project-C-WPF>

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Маккефри, Д. "Learning SQL Server Management Studio" - Wiley, 2018.

2. Майклс, Р. "Professional C# 7 and .NET Core 2.0" - Wrox, 2018.

Интернет-ресурсы

3. Microsoft Documentation [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/sql/ssms/>  
 4. CodeProject [Электронный ресурс] – Режим доступа: <https://www.codeproject.com/>

5. Stack Overflow [Электронный ресурс] – Режим доступа: <https://stackoverflow.com/>

6. SQLShack [Электронный ресурс] – Режим доступа: <https://www.sqlshack.com/>

7. SQLServerCentral [Электронный ресурс] – Режим доступа: <https://www.sqlservercentral.com/>

8. C# Corner [Электронный ресурс] – Режим доступа: <https://www.c-sharpcorner.com/>

9. Pluralsight [Электронный ресурс] – Режим доступа: <https://www.pluralsight.com/>

10. Udemy [Электронный ресурс] – Режим доступа: <https://www.udemy.com/>